Exercise 3:

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithms**  **Memory partitions** | **First-fit** | **Best-fit** | **Worst-fit** |
| **300 KB** | 115 KB | --- | --- |
| **600 KB** | 500 KB | 500 KB | 358 KB |
| **350 KB** | 200 KB | --- | 200 KB |
| **200 KB** | --- | 200 KB | --- |
| **750 KB** | 358 KB  375 KB | 358 KB  375 KB | 115 KB  500 KB |
| **125 KB** | --- | 115 KB | --- |
| Available memory | 777 KB | 777 KB | 1102 KB |

The most efficient algorithm in terms of using memory is the Best-fit because its internal fragmentation is smaller than the others. Then comes First-fit, though its available memory is the same as Best-fit but internal fragmentation is bigger. The worst algorithm is Worst-fit because it creates an external fragmentation which that cannot fit the last process (375 KB) to any partitions.

Exercise 4:

|  |  |  |
| --- | --- | --- |
|  | **ADVANTAGES** | **DISADVANTAGES** |
| **FIRST FIT** | * Simple, fast, tends to produce larger free blocks. | * The remaining unused memory after allocation is wasted & External fragmentation |
| **BEST FIT** | * Memory utilized | * Slower in operations, may have tiny useless fragments |
| **WORST FIT** | * Reduce the rate of production of small gaps & works best if allocations are of medium sizes | * External fragmentation & tends to break large free blocks that large partitions cannot be allocated |